

客户 / 服务器模式在医院 信息系统中的实践

薛万国 (解放军总医院计算机室)

传统的以主机为中心的集中处理方式下,主机的能力决定一切。许多应用需求,如要求快速响应的联机应用、要求有方便灵活的用户界面的应用、含有图形图像处理的应用等,受主机及终端能力的限制,或者难以实现,或者费用昂贵。个人机及工作站的出现,使应用程序的面目焕然一新。快速的反应能力、方便的用户接口、丰富的工具都是前所未有的。然而,由高度集中走向高度分散的信息孤岛在信息系统中也是行不通的。医院信息系统的高度共享的特点使得我们必须将这些孤岛构成一个整体。在这种资源分布的情况下,优化资源配置,提高系统整体性能,降低资源消耗浪费即降低系统造价的最佳方法就是采用客户 / 服务器模式。如果说,信息技术的发展为规模缩小化提供了可能话,那么客户 / 服务器模式为缩小化提供了最优途径。

在获得好处的同时我们也应看到,信息由高度集中发展到相对自治,由自主工作发展到协同工作,必将给开发人员及信息系统的管理者带来许多新问题。这些问题包括:系统的可靠性、数据的安全性、一致性等等。只有将这些问题解决了,客户 / 服务器模式才能顺利实施,其优势才能体现。

我们医院的信息系统过去是典型的集中式结构,从92年开始向客户 / 服务器模式转化。我们进行了两个方面的工作:一是建造了一个客户 / 服务器模式的信息系统支撑环境;二是在该环境下开发了若干客户 / 服务器模式的医院应用系统。在这两项工作中,我们获得了对客户 / 服务结构的较为直接的认识与体会。客户 / 服务器给信息系统带来的究竟是什么?这是许多用户所关心的,在一些问题上也有不同看法。本文将在介绍这方面所进行的工作的基础上,结合应用经验,针对实施客户 / 服务器结构的几个问题谈一些自己的认识。

一、客户 / 服务器结构

尽管对客户 / 服务器模式有各种各样的理解与定义,然而,就其实质来说,客户 / 服务器模式体现的是“合理分工,协作处理”,即一个任务由合理分布在客户机上的前端和服务器上的后端协作完成。

不同类型的客户 / 服务器模式的不同之处在于协作处理内容的不同,特别是服务类型或级别的不同。对于一个信息系统来说,其处理的核心是数据库操作。只有当服务器提供数据库服务时,才能使客户机有效地共享服务器强大的处理能力,才能够简化前端应用程序对数据库的操作过程,才能减少网络流量。而对信息系统来说,文件服务不能解决数据库操作问题,所有的数据库操作都要由本地客户机承担;而智能终端结构使服务器承担了几乎所有处理,且与应用的相关性极大。上述两种情况均不能实现“合理分工”,是处理分布的两种极端。综上所述,站在信息系统的角度,客户 / 服务器之间应以数据库请求 / 服务为基础。对医院信息系统也是如此。本文后面所指的客户 / 服务器模式应属此类。



图 1 客户机 / 服务器模式开发工具的三个部分

开发客户 / 服务器模式的信息系统,必须有相应的开发工具及运行环境的支持。一个客户 / 服务器应用开发工具应有三个要素:数据库服务、应用程序编程接口、网络通讯。其中数据库服务位于服务器方,构成应用程序的协作方,负责服务器上的数据库处理。由于 SQL 语句所具备的强大完备的表达能力,客户机通常以 SQL 语句作为请求描述语言,这就要求数据库服务具备 SQL 处

理能力。此时,数据库服务又称为 SQL 服务器。应用程序编程接口是应用程序唤醒服务程序进行协作处理的接口点,它负责向服务程序发出处理请求,并接收处理结果。这一接口多以过程调用的形式提供给应用程序负责传递请求与结果。这三部分的结构关系见图 1。

二、客户 / 服务器模式信息系统支撑环境的开发

我们开发的客户 / 服务器模式信息系统支撑环境是基于这样的思想:充分利用现有的软硬件资源,应具有良好的可扩充性,具有较低的价格。我们设计了图 2.所示的体系结构。

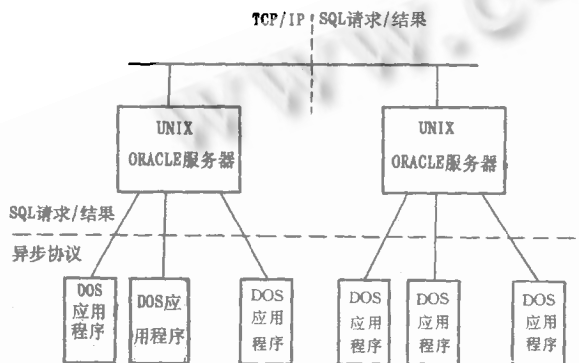


图 2 客户机 / 服务器模式开发工具的三个部分

目前许多用户拥有微机,其中绝大多数为 286 档次的微机,这些微机运行 DOS 操作系统,对单个用户具备了较好的处理及存储能力,我们用其作为客户机;为了共享数据,用 386 或 486 的高配置微机作为服务器,运行 UNIX 操作系统;在客户机与服务器之间通过异步通讯进行点一点连接,服务器通过多用户卡可以连接多台客户机;根据需要可以配置多台服务器,服务器之间通过以太网及 TCP/IP 协议进行连接;在服务器上运行 ORACLE 数据库。

在这样的结构下,用户机既可以以客户机方式与服务器联接,也可以以终端方式直接连在服务器上,因为服务器实际上就是一台运行 UNIX 的主机。整个系统不仅可以通过升级服务器得到规模上的纵向扩充,也可以通过增加服务器得到规模上的横向扩充。信息系统的建

设可以逐步进行,可以不断享有硬件高速发展带来的更高性能价格比的最新产品,前期的投资将得到保护。由于 UNIX 与 TCP/IP 良好的开放性和可连接性,使得服务器扩充及与外界的联系非常容易。每一台客户机不管直接连在哪台服务器上,都可以通过自己的服务器与别的服务器共享数据及处理资源。对那些通讯量大的应用如图像传递,也可以将客户机通过以太网直接与服务器相连。因此整个结构的配置非常灵活,同时不失开放性。

下面通过客户 / 服务器的三个要素,简单说明我们的支撑环境在设计及实现方面的特点,更加具体的细节,请参阅文献[3]和[4]。软件结构见图 3。

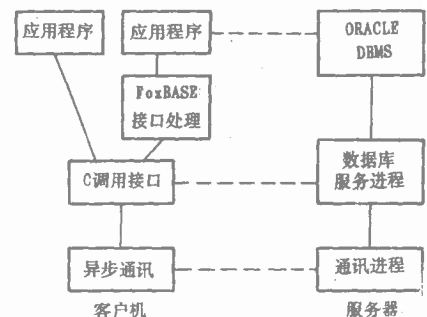


图 3 支撑环境的软件结构

1.数据库服务

我们在 UNIX 服务器上开发了基于 SQL 的数据库服务程序,它依靠 ORACLE 的动态 SQL 语句处理功能将客户机送来的 SQL 语句提交 ORACLE 执行,并将执行结果打包回送。它不仅支持所有的 SQL 语句,而且针对应用程序的编程需要,实现了支持同时激活多条 SELECT 语句,允许逐条提取满足查询条件的记录等功能。

2.编程接口

我们不仅在客户机方提供 C 语言的编程接口,而且针对开发信息系统的需要,实现了 FoxBASE 的调用接口。这样,应用程序可以在客户机上用 FoxBASE 开发用户接口,并且可以使用本地数据库,同时可以在程序中以 SQL 语句操作服务器上的数据库,并将操作结果返回到 FoxBASE 变量中。支撑环境提供在 ORACLE 数据类型与 FoxBASE 和 C 语言类型间自动转换,这一功

能使得异种数据库管理系统很好地连接起来,从而可以充分利用 FoxBASE 在开发信息系统中的长处。

3.网络通讯

我们之所以采用异步通讯是基于这样的考虑:高速网络如以太网,速度快、固然好,但因院区范围大,造价太高,而异步通讯造价则相对低得多;二是采用客户/服务器模式之后,由于客户机与服务器之间仅传送 SQL 语句与服务结果,与基于文件服务的信息系统相比,数据传输量很小,采用异步通讯完全可以满足速度上的需要。(如果在 10 万条病人记录中顺序查找一条的话,客户/服务器模式下网络上仅传送一条 SELECT 语句及一条记录,而文件服务方式下最多可能要传送 10 万条记录。)这是客户/服务器模式带来的网络方面的节省。当然,对于数据传输量大的应用,也可以直接通过以太网与服务器相连接。

基于异步通讯,我们设计开发了异步通讯协议。它具有错误检测、出错重发的能力,可以传输二进制数据。协议的所有处理对高层透明,可以确保数据在客户机与服务程序之间正确无误地传输。

这一支撑环境从本质上具备了前述的客户机/服务器模式的准则。

三、客户/服务器模式应用系统 开发经验谈

我们在上述支撑环境上开发了若干个应用系统并已投入使用。目前我们采用一台 AST PSE4/33 作服务器(486/33 CPU, 16 M 内存, 1 G 硬盘), 连接 40 个前端用户(多为 286 兼容机), 在整体性能及造价上均得到了较好的效果。在开发及实施应用系统的过程中,我们对客户/服务器模式的信息系统有了一个更全面的认识。下面,就系统的性能、造价、系统的安全性、数据一致性等几个用户关注的焦点问题来谈一下体会。

1.性能

由于应用在客户机与服务器间合理分布,客户机与服务器的各自优势都得到了发挥。与集中式系统的终端相比,用户界面的反应速度更快,更容易使用;与基于文件服务的网络相比,充分利用了高档服务器的处理能力。也就是说如果客户机是 286 微机,服务器是 486,则在进行数据库查询时,286 获得的是 486 的处理速度,而

在文件服务方式下,获得的是 286 速度。通过以下数据可以更加量化的方式说明问题。

我们曾经作过这样的实验:在 4 M 内存的 486 服务器上装有 10 万条记录的数据库,分别以 16 MHz 主频及 12 MHz 主频的 286 机作客户机,对服务器上的数据库作顺序检索。客户/服务器模式支撑环境采用异步通讯,网络 FoxBASE 采用 10 M 以太网。测试结果如表 1 所示。

表 1 支撑环境和网络 FoxBase 查询性能对比

	16 MHz 286	12 MHz 286
C/S 支撑环境	45 秒	45 秒
网络 FoxBase	6 分 29 秒	10 分 22 秒

在目前我们运行的信息系统中,通过异步口连接了 40 台客户机,在这些用户中,同时联机使用的约 20 台左右,通过对 CPU 负载的跟踪,在每天上午最繁忙时,CPU 有 70-80% 的时间为空闲,这是包括我们在内的许多人意料不到的。与终端方式相比,这是客户机分担主机负载的结果。

通过另一例子说明对用户的响应速度。在医院信息系统中,门诊病人划价收费是典型的实时事务处理。无论是从数据录入量,还是对响应速度的苛刻要求,在其它系统中是不多见的。然而由于我们采用了客户机/服务器模式,由客户机承担了对速度要求高的划价工作,服务器承担数据的存储及结账统计工作。使得每一种药品或收费项目的录入划价,对熟练收款员无等待感觉,响应时间趋于 0,完全满足了速度要求。每个窗口每天录入 200 张以上处方,而对主机 CPU 的消耗,一个窗口不过 3 秒钟左右,这样的应用需求在终端方式下是难以实现的。

2.造价

一个信息系统的造价应包括运行环境的造价以及开发应用软件的代价两个部分。从运行环境来看,同样性能的微机(也许不应该称为微机)要比专有系统结构的主机在价格上低得多,这已成为共识。另外,从系统的扩充及维护费用来看,使用开放系统及微机技术将比专有系统更节省。这是开放系统本身带来的好处。另一重要因素,即在同等应用规模的条件下,达到同样的性能,采

用客户/服务器模式对主机资源的消耗大为降低,过去由主机完成的工作现在可由高档微机完成而不会降低系统性能,从而可以进一步减少投入。这一点从性能方面的分析及实验可以看出。这是客户/服务器模式对资源合理配置、合理利用带来的好处。

至于开发应用系统的费用,国内外用户对实施客户/服务器模式有不同的看法。根据我们的经验,这一问题我们认为应从两个方面来看。一方面,客户/服务器模式与集中系统相比,确实增加了开发的难度及工作量,设计应用系统时要考虑的因素增加了。增加的工作量视应用的不同而不同,高者可能达到 50%或更多,这是不利的一面;另一方面,由于客户机/服务器模式的实施,使得程序员可以利用客户机上丰富的开发工具来开发应用,特别是开发用户界面更是如此。与终端方式相比,可以以更少的工作量开发出更好的界面,从而提高编程效率。这种提高可能是成倍的。这是有利的一面。

3. 系统的安全性

系统的安全性是信息系统的管理者及用户关心的重要问题。处理及数据的分布在带给我们好处的同时,也给信息系统增加了数据破坏或丢失的机会,这正是事物的正反两个方面。我们的做法是,通过将重要数据集中存放等措施,将这一影响减小到最低程度。在医院信息系统中,有许多信息,象病历,是高度共享的。这一类重要的信息以服务器存放为主,客户机仅起到录入界面及数据校验的作用。从服务器方面看,所有关系全局的重要信息都保存在服务器,只要保证了服务器的数据安全,整个系统的安全就有了保证。事实上,在我们的信息系统支撑环境中,从服务器方看,它与传统的 UNIX 主机并没有明显区别。通过 UNIX 的安全机制使共享数据得到有效保护。同时,对那些保存在客户机上的信息,在设计开发应用程序时要提供备份及恢复手段,由用户定期执行数据备份工作,当然这增加了用户的工作,但进一步保证了数据的安全。采取这些措施之后,在我们的信息系统运行一年里,并没有出现数据丢失而不能恢复的灾难。

4. 数据的一致性

由于数据的分布存放,特别是为了提高系统效率,某些信息(比如收费系统中的价表)可能在客户机与服务器

上重复存放(这是客户/服务器常采用的手段),这就带来了数据的一致性问题。对此我们采取的方法是,将那些更新频率大,且需共享的数据集中存放,避免不一致。那些不常更新或实时性要求不高的数据,如价表、各种编码字典等,为提高效率可以重复存放,但在应用系统设计时提供定期更新手段以达到异步的一致性。比如我们在收费系统中,如果价格管理员通过程序修改了主机上的价表,正在联机工作的收款机(客户机)的价表并不立即更新,而是等到收款机再次联机工作时,程序自动检查主机价表,并与主机取得一致。这一更新延迟一般最多为半天,对收费系统来说,这是允许的。总之,只要我们认识到这一问题并认真对待,数据一致性是可以解决的。

四、结束语

客户/服务器模式是信息系统的—个发展潮流,越来越多的用户,越来越多的厂商正在加入这一行列,医院日益发展的应用需要正在呼唤这一技术。我们立足于现有资源开发了客户/服务器模式的信息系统支撑环境,并在其上开发了一系列医院应用系统。从客户/服务器模式的共同特征来看,从应用实践中,我们感到它确实带给我们投资上的节省,给用户带来了更高的性能及更加易于使用的界面,也为图形图像等更进一步的应用提供了方便。同时,客户/服务器模式也对我们的开发工作提出了新的要求,我们应该充分认识到客户/服务器模式带来的系统安全性、数据一致性问题,并在设计及实施中采取措施,加以解决。只要措施得当,这些问题是可以控制的。

参考文献:[1]. 龚波,“浅议客户机/服务器(Client/Server)计算环境”,国际电子报,1993年,第39期。

[2]. 叶皓,“Client/Server及其推广应用的紧迫性”,计算机世界报,1993年,第38期,第9页。

[3]. 任连仲等,“一种集中—分布信息系统结构及其支撑环境”,计算机系统应用,1992年,第1期,第13页。

[4]. 薛万国等,“客户/服务器模式下SQL服务功能的设计与实现”,计算机系统应用,1992年,第2期,第2页。